Functional Object Descriptors for Human Activity Modeling

Alessandro Pieropan Carl Henrik Ek Hedvig Kjellström

***Abstract*— The ability to learn from human demonstration is essential for robots in human environments. The activity models that the robot builds from observation must take both the human motion and the objects involved into account. Object models designed for this purpose should reflect the role of the object in the activity – its function, or affordances. The main contribution of this paper is to represent object directly in terms of their interaction with human hands, rather than in terms of appearance. This enables the direct representation of object affordances/function, while being robust to intra-class differences in appearance. Object hypotheses are first extracted from a video sequence as tracks of associated image segments. The object hypotheses are encoded as strings, where the vocabulary corresponds to different types of interaction with human hands. The similarity between two such object descriptors can be measured using a string kernel. Experiments show these functional descriptors to capture differences and similarities in object affordances/function that are not represented by appearance.**

I. INTRODUCTION

An important capability for robots in human environments is to interpret and learn from perceived human activity. This is known as robot learning from demonstration [1] or imitation learning [2]. Imitation here means observing an action and its effect on the world, and performing an action that has the same effect [3]. Thus, a human activity modeling method intended for imitation learning should represent human activity not only in terms of human motion, but also in terms of objects involved in the activity [4].

A common approach to employing such object context in human activity recognition [5], [6], [7] is to learn statistical dependencies between appearance based object classifications and human action classifications in a body of training data. Objects are categorized into semantic classes such as `cup`, `cricket-bat`, etc., and represented in terms of visual features; for an overview of object recognition research, see [8]. When observing a new activity example, the object context helps enhancing the human action classification, and human motion context gives information about the function of the object in the observed activity.

The assumption underlying [5], [6], [7] is that the semantic class of the objects involved is directly correlated with the activity. This is partly true, as semantic classes are often coupled to function – `cups` are, e.g., generally used to contain hot liquid, `glasses` are used for cold liquid, while `plates` are generally used for non-liquid food, which means that it is appropriate for a robot to reason about objects in terms of `cups`, `glasses` and `plates` when learning to serve food and drink.

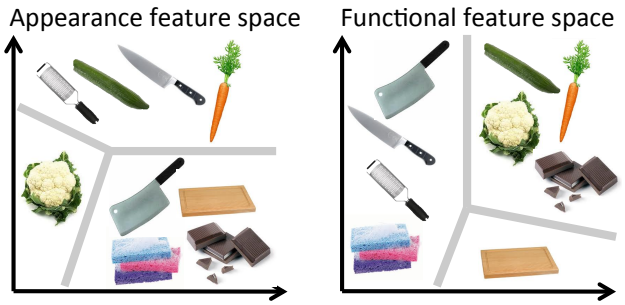


Fig. 1. Appearance based vs. function based object modeling. *Left:* If objects are characterized by appearance (shape, color, texture) features, a set of objects might be grouped into elongated, round and square. *Right:* Using features reflecting the object function in human activity, the objects might instead be grouped into tools, ingredients and support areas.

However, for robot reasoning and learning about activity, the *core aspect of objects is the function itself* – the semantic class of an object is merely an intermediate representation of functionality. (This is in contrast to other applications of object recognition [8], such as image retrieval, where the semantic class is the core representation.) This is in accordance with Gibson's *affordance theory* [9], which states that humans and animals recognize objects directly in terms of their function; what affordances they allow for the task at hand.

To a certain extent, object function/affordances can be understood directly from visual information [10], [11]. The robot's learning of function-appearance correspondences can be further guided by human demonstration [12], [13] or by letting the robot perform actions on the object using its own actuators [14], [15], [3]. However, there are functional properties that can not be observed visually in a direct way, such as temperature, flexibility and weight. Furthermore, there are functional object classes with a very wide variety in appearance, e.g., chairs [16]. Fig. 1 shows an example of a domain where the functional object classes relevant for the activity are virtually impossible to discern from object appearance only.

In this paper we go one step further: Objects are here *characterized directly in terms of how they normally interact with humans*. In other words, we directly represent objects in terms of function/affordances, without taking the visual appearance into account. This is similar in spirit to recent work [17], [16], [18], [19] in which scene regions with certain affordances, or function, are detected by observing (real or simulated) humans interacting with them.

This research has been supported by the EU through TOMSY, IST-FP7-Collaborative Project-270436, and the Swedish Research Council (VR).

The authors are with CVAP/CAS, KTH, Stockholm, Sweden, `pieropan,chek,hedvig@kth.se`.

The data on which the reasoning is performed are RGB-D (Kinect) video sequences showing a human in interaction with objects. Object hypothesis tracks are extracted in an unsupervised way from the video (Section III).

The human in the scene is tracked using the tracker built into the Kinect [20], rendering the 3D human hand positions. Using this information, all object hypotheses can in each frame be labeled according to their relation to the hands: Whether the object is close to a hand, being held by a hand, approached by a hand, and so on. The result is a string of labels, one label for each frame over the duration of the object hypothesis track. This representation is independent of the appearance of the object, and reflects instead the nature of interaction between a human and the object.

The central assumption in the present work is that these *object string descriptors encode the functionality of objects in human activity*: A `tool` is frequently in the human hand, a `cutting-board` is frequently close to a human hand but seldom held, while a `container` is seldom approached by hands. Thus, the statistics of an object string are correlated to the functionality of the object.

To be able to reason about objects in terms of their string representation, a measure of similarity must be defined. A standard distance measure, e.g., a Euclidean distance, will not do, since the strings themselves vary between members of the same class. Consider e.g. two observations of hammers, where one is picked up in the beginning of the activity, but the other is not used until the very end. Their state sequences will then be completely disparate. However, the *statistics* about how often and for how long they are in the human hand are similar. Thus, the similarity measure should capture similarities in statistics, while ignoring the specific order of labels in the strings.

The string correspondence problem has a direct parallel in text comparison, where similarity is defined in terms of local structures (single words, bi-words, tri-words) rather than global structure. We therefore define the similarity between two object strings in terms of a string kernel [21], originally designed for text comparison. Section IV expands on the extraction of object string descriptors, and the similarity between them.

The contributions of this paper are: I) the idea of *representing object directly in terms of their interaction with human hands*, rather than in terms of appearance; II) the idea of *encoding the objects as strings*, where the vocabulary corresponds to different types of interaction with human hands, and *measuring their similarity using a string kernel*.

These functional object string descriptors can, together with their similarity measure, be employed in modeling and recognition. To evaluate the descriptive power of the functional representation, we extract string descriptors from a body of data consisting of object hypotheses from a realistic scenario. The performance of our descriptors is compared to a baseline of appearance-based descriptors (SIFT bag-of-words) in terms of classification using an SVM. The classification accuracy using our descriptors is 92%, compared to 64% for the appearance descriptors (Section V).

II. RELATED WORK

The concept of affordances [9] has come in focus lately within the Cognitive Vision and Robotics communities. While many other papers on affordances, e.g. [22], [23], [24], [13], concentrate on robotic grasping, we here focus on more composite, higher-level actions, which typically involve grasping as a sub-component.

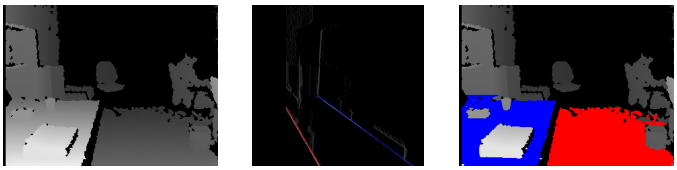
As described above, object function or affordances are often modeled in terms of appearance, and objects are categorized into semantic classes, not necessarily with a one-to-one mapping to functional/affordance classes [25], [5], [6], [26], [27], [7], [28], [29]. In contrast, we describe objects directly in terms of affordances, in other words, in terms of their function in human activity.

The embodied/cognitive vision approach to affordance learning consists of an agent acting upon objects in the environment and observing the reaction. In [14], [30], a robot pushes, pokes, pulls and grasps objects with its end-effector, thereby extracting object hypotheses and learning about rolling, sliding etc. Montesano et al. [3] notes that an affordance can be described by the three interdependent entities of action, object, and effect. A robot first learns a set of affordances by exploration of the environment using preprogrammed basic motor skills. It can then imitate a human action, not by mimicking the action itself, but rather observing the effect and then selecting its own action that will have the same effect on the current object. In the work presented here we will extend this and also learn the object affordances from human demonstration.

To a certain degree, affordances can be observed in images. In recent work on grasping [22], [23], [24], [13], relations between visual cues and grasping affordances are learned from training data. In [13], object grasping areas are extracted from short videos of humans interacting with the objects, while in [22], [24] the grasping affordances are associated with appearance, learned from a large set of 2D object views labeled with grasping points.

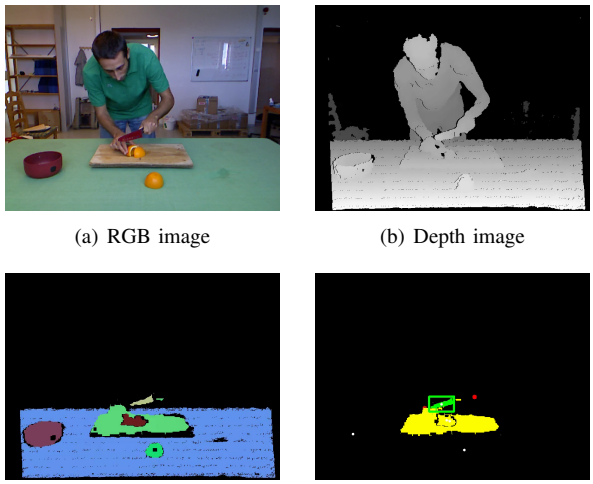
Early work on functional object recognition [10], [31] can be seen as a first step towards observing more general classes of object affordances in images. Objects are there modeled in terms of their functional parts, e.g., `handle` or `hammer-head` [10], or by reasoning about shape in association to function [31]. More recently, Gall et al. [12] cluster objects into functional classes based on observations of human activities performed upon them. Like our method, no pre-trained appearance based classifiers are needed. However, while we extract explicit functional descriptors, they cluster objects based on the associated human motion. Our explicit descriptors makes it possible to represent object affordances/function in a larger activity modeling framework, and also to integrate the functional representation with one of object appearance (Section VI). Furthermore, we regard activity on a longer time scale and in terms of spatial relation between hands and objects, while they focus on individual human actions and in terms of upper body motion.

The recent development in depth sensing has introduced



(a) Depth map (b) Hough image with detected lines (c) Depth map with detected planes

Fig. 2. Detection of horizontal planes. (a) Depth map $\mathscr{D}_t$. (b) Hough image $\mathscr{H}_t$ with detected lines. (c) Depth map $\mathscr{D}_t$ with back-projected detected horizontal planes.



(a) RGB image (b) Depth image (c) Segmentation of object hypotheses (d) Annotation of object-hand relation

Fig. 3. Extraction and annotation of video. (a) RGB image. (b) Depth image. (c) Segmentation of object hypotheses. (d) Annotation of object-hand notation. Object color coding: • = **Idle**, • = **Close to**, • = **in Use**. The white dots are object centroids while the red dots are hand (wrist) positions. *More examples are shown in* `www.csc.kth.se/~pieropan/videos/ICRA2013.avi`.

new possibilities for functional recognition in images. In [11], functional parts of chairs are recognized from depth images, while [16] take a different approach to the same problem, simulating humans performing sitting actions in a depth image of the scene, detecting sitting affordances. In [18], [32], real humans are instead observed in interaction with a scene; their interaction with the scene is used to detect affordances in a similar manner. Like in our work, appearance is not taken into account.

Our object string representation relates to recent work in visual activity recognition [33], [34] where activity is represented in terms of strings, encoding the interaction of image areas over the duration of the activity. Like our object representation, the activity string representation is completely independent of object appearance, only by the interaction of different objects.

III. EXTRACTION OF OBJECT HYPOTHESES

The basis of the analysis is a set of object hypotheses in the form of tracks of image segments over time. The actual segmentation and tracking of object hypotheses is not a focus of this paper, but is described here for completeness.

We make the assumption (valid in indoor environments) that objects involved in human activity are located on horizontal support areas (tables, floors, shelves) in the beginning of the activity. They can later be picked up, which means that they are in the human hands. These assumptions can be used to constrain the object hypothesis detection in the video.

Firstly, horizontal support areas are detected at time t_0. Following the approach of [35], the pixels of the depth map $\mathscr{D}_t$ (Fig. 2(a)) are represented as a cloud of 3D points $(x, y, \mathscr{D}_t(x,y))$. A Hough image $\mathscr{H}_t$ is created by marginalizing the point cloud over x and computing a point histogram $\mathscr{H}(z,y) = \sum_x (\mathscr{D}_t(x,y) = z)$. Lines in $\mathscr{H}_t$ correspond to planes that are horizontal in the x direction, and are detected using a Hough transform technique [36] (Fig. 2(b)). By back-projecting the detections to the original $(x, y, \mathscr{D}_t(x,y))$ points in the cloud, the planes can be detected in the depth image $\mathscr{D}_t$ (Fig. 2(c)). This assumes that the camera is oriented horizontally, i.e., that horizontal lines in the image are also horizontal in the world.

Object hypotheses are then defined as areas connected to the planes in the image, but with depth and/or color deviating from the planes. Such areas are at t_0 segmented out from the plane regions in the image using the connected component algorithm [37].

To maintain and associate object hypotheses over time, HSV histograms are extracted from each object segment at time t_0, and the object mask is maintained over the course of the activity by foreground segmentation using this histogram (Fig. 3(c)). A pyramid KLT tracker [38] is initialized on each segmented hypothesis at t_0, and used to detect if the object is stationary or moving. If stationary, the bounding box of the object is assumed unchanged between frames t and $t+1$. If the object has moved, the tracker is used to estimate the new object bounding box at $t+1$. The foreground segmentation at $t+1$ is then carried out within the estimated bounding box.

Using this approach, a set of object hypothesis tracks – time sequences where each time step contains object mask, size and position of the object bounding box, and RGB-D values for each pixel in the bounding box – can be extracted from the RGB-D video of an activity.

We will now proceed to explain the main contribution of this paper, by presenting how objects can be presented and encoded not, as traditionally, using their appearance but from their functional characteristics.

IV. FUNCTIONAL OBJECT REPRESENTATION

In the previous section we outlined an approach that generates object hypotheses from dynamic scenes. Given a hypothesis we now wish to represent each object in a manner corresponding to the functional characteristics of the object. The argument we make in this paper is that such information can be provided from the characteristics of a human's interaction with the object. In this section we

will outline how such interaction patterns can be robustly represented.

From each object hypothesis the spatial location can be extracted as a temporal trajectory. Further, we estimate the position of the hands of the human interacting with the objects using the tracker built into the Kinect [20]. Clearly, the relative position between object and the hand provides relevant information relating to the functional properties of an object. To that end we will let the temporal signature of the relative position between the hands and the object constitute our object representation, see Fig 3(d). Rather than using a continuous state-space we discretize the space into five different states as follows: let d_t be the Euclidean 3D distance from the closest hand to the closest point on a segment s at time t. The possible states x_t of a segment s at time t are:

$x_t =$ I (Idle): $d_t \geq D_1$
$x_t =$ A (hand Approaching): $d_t < D_1$ & $d_{t-1} > d_t$
$x_t =$ L (hand Leaving): $d_t < D_1$ & $d_{t-1} < d_t$
$x_t =$ C (Close to): $d_t < D_2$
$x_t =$ U (in Use): $d_t < D_2$ & segment s is moving

where $D_1 > D_2$ are distance thresholds.

Given the discretized representation above, each object hypothesis can be represented as a sequence of states over time; we will refer to this as a string. Fig. 5 shows an example representation for two object hypotheses over five time steps. At present, the string language {I, A, L, C, U} is manually defined; future work (Section VI-A) includes learning, both the vocabulary and the classification of data into words, from the data.

As explained in the Introduction, we argue that the functional characteristics of an object is encoded in the string which describes the interaction patterns between the human and the object. Knives and other tools move for longer times than ingredients, who move during some activities (putting them on a cutting board or in a bowl) but not during others (cutting them). Cutting boards seldom move but the hands are often close to them. However, within the same functional category we expect there to be variations both in terms of duration, and permutations of order, rendering strings of different length in several different arrangements. In addition, due to uncertainty in the labeling process we are likely to produce strings contaminated by noise. Techniques for dealing with noise for vectorial data are very well developed, with examples such as Support Vector Machines for classification and Gaussian Process models for regression. Much less work exists for dealing with sequential information such as the strings generated from our object hypotheses. To that end, we take the approach of first converting the strings into a vectorial representation that facilitates application of robust learning techniques. The naïve approach would be to simply consider the string as a vector and compare it using a regular metric. However, as strings potentially have different length, this will not be possible. Further, even if possible, such a representation would not have any of the characteristics that we seek, such as sensitivity to order and permutation. A much more principled way of transferring strings to a vector

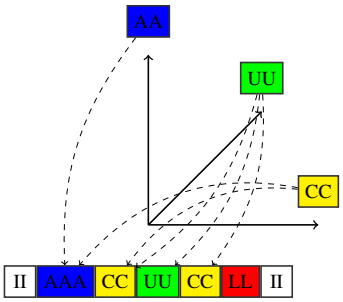


Fig. 4. The above figure describes the feature space induced by the string kernel used to describe the data. Below is a string of states generated from the segmentation system and above is a couple of basis functions. Each part of the string containing the basis substring generates a non-zero projection corresponding to the length of the match. As an example the first part of the string, **IIAAACC** will have a non-zero contribution to **AA** and with parts (**II**)**AA**(**ACC**), (**II**)**A**(**A**)**A**(**CC**) and (**IIA**)**AA**(**CC**) where the bracket shows states not part of the mapping.

space is to exploit the kernel trick, by implicitly defining the space of strings through a kernel function. One such technique has been proposed by [21] and is referred to as a string kernel; this is what we will use here.

The string kernel, in contrary to most other kernel based approaches, explicitly defines its feature space but then uses an efficient kernel function to represent the mapping from the string space. In specific, the feature space is spanned by each permutation of the states up to length K, which means that the dimensionality of the space is bounded. The projection of a string onto a basis vector depends on the overlap between the two strings where a spatially bigger overlap will result in a less orthogonal string. This will generate a valid kernel inducing a feature space where the geometrical configuration will gracefully reflect the similarity of the strings. In Fig. 4, a schematic explanation of the string kernel feature space from the state sequences is shown.

V. Results

For the application in focus here, robot learning from demonstration, the human is involved in an activity with the intention to showing the robot how this activity is performed. Throughout this paper, we use the example of preparing food.

Since there are no existing datasets comprising human activity captured with RGB-D, with human pose data, we evaluate the method on a dataset collected specifically for this paper. We will make this dataset publicly available so that our results can be compared to others.

The dataset includes 40 RGB-D videos captured in an indoor environment, involving one subject cutting vegetables, fruit and bread, placing them in bowls and pans (Fig. 6). The set of objects includes 4 different knives, 9 ingredients, 3 cutting boards and 4 containers. The activities are performed in a natural manner without clear separations between the different actions. The video data are captured in 20 Hz, with a resolution of 640×480 pixels and average length of 50 seconds. Human pose is extracted using the method included with the Kinect sensor. [20].

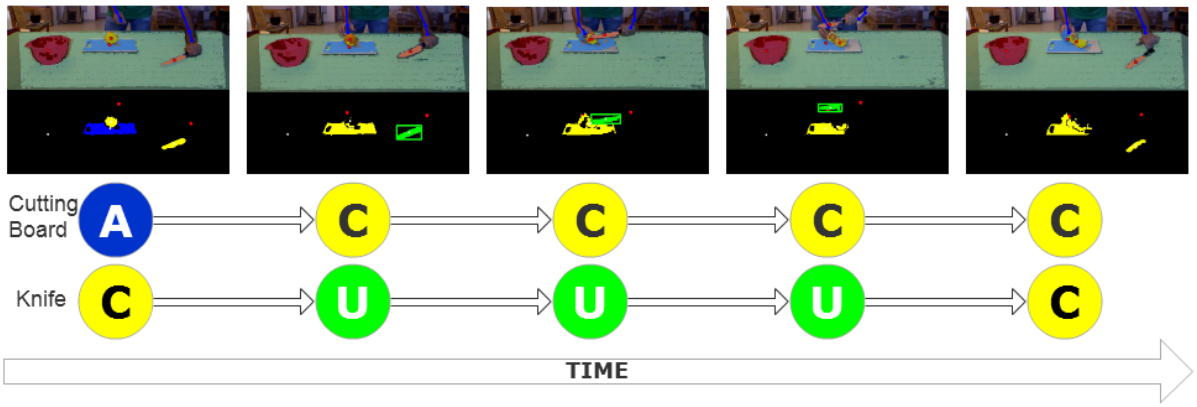


Fig. 5. Example of state transitions for two different segmented areas over five time steps in a sequence. The cutting board is first **Approached** and then **Close to** the hand. The knife is first **Close to** a hand, then **in Use** when held by the human and used on an ingredient and finally **Close to** a hand when it is left on the table.

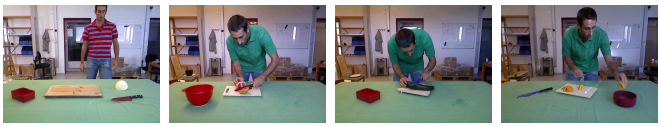


Fig. 6. Examples from the dataset, containing kitchen activities involving different tools, ingredients, containers and cutting boards.

In this scenario, the objects can be categorized into four functional classes: **Tools**, objects used to perform actions on other objects, e.g., knifes. **Ingredients**, objects acted upon by tools, e.g., fruit. **Support areas**, objects used as support, e.g., cutting boards. **Containers**, objects that can contain other objects, e.g., bowls. Object hypotheses are extracted from the video using the procedure described in Section III, and the string descriptor of each hypothesis are found as described in Section IV. The hypotheses are then labeled manually with the correct functional class, to enable ground truth comparison.

a) Functional descriptors and string kernel: As detailed in Section IV, the similarity between two object strings $\mathbf{x}_1$ and $\mathbf{x}_2$ can be measured using the string kernel as $k_{\mathbf{string}}(\mathbf{x}_1,\mathbf{x}_2)$. The pairwise similarities using this measure are computed between all pairs of object hypotheses in the data, with the parameter settings $K = 2,3,5,7$. A K smaller than that equals Baseline 1 below, and a larger K has empirically been found to increase the computational effort without any improvement in performance.

Fig. 7(a) shows the pairwise similarities for $K = 3$. The rows of the matrix are ordered according to the ground truth label of the object hypotheses, in the order of **Tools**, **Ingredients**, **Support areas**, and **Containers**. It is apparent from the matrix that the within-class similarities are higher than the between-class similarities, i.e., that there are clear clusters in the functional feature space, each cluster corresponding to a certain functional class.

b) Baseline 1: Functional descriptors, 0^{th} order statistics: The most straight-forward way of representing a string $\mathbf{x}$ is the Bag of Words (BoW) approach – to collect the statistics of each of the 5 words (I, A, L, C, U) into a histogram $\mathbf{h}$, normalized to sum to 1. This histogram will be a very compact, 5D representation of the string $\mathbf{x}$. $\mathbf{h}$ is the 0^{th} order statistics of $\mathbf{x}$, since it ignores all information about word order. The similarity between two normalized histograms is measured with the intersection kernel

$$k_{\mathbf{intersect}}(\mathbf{h}_1,\mathbf{h}_2) = \sum_{i=1}^{n} \min(h_{i,1},h_{i,2}) \quad (1)$$

where the dimensionality $n = 5$ in this case.

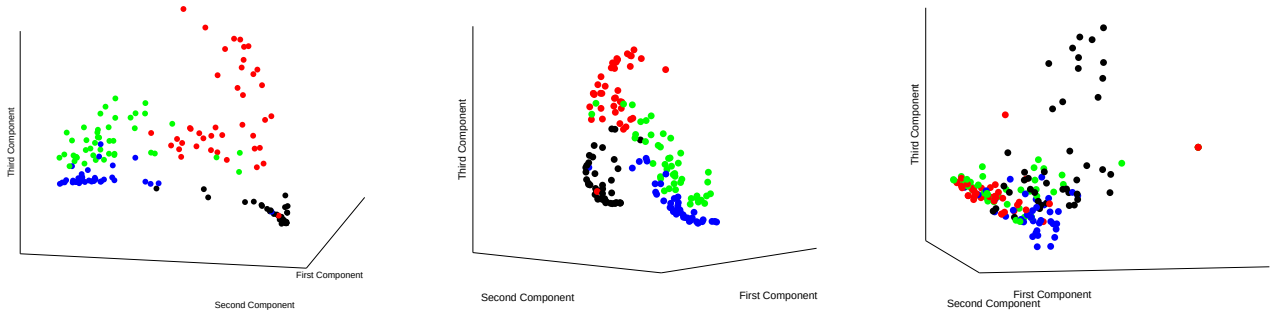
As a baseline to the string kernel performance on the object strings, the pairwise similarities using the BoW representation with the intersection kernel are extracted as shown in Fig. 7(b). The differences between Fig. 7(a) and Fig. 7(b) are that the string kernel performs slightly better than the bow representation, we believe that depends on the lack of multiple usages of objects. We believe that in a more complex scenario with repeating actions the temporal information encoded in the string kernel will become more significant.

c) Baseline 2: Appearance similarity measure: We also compare the functional string representation to a baseline of a standard appearance representation, SIFT [39] BoW.

For each object hypothesis, SIFT features are extracted from the segmentation mask in all frames with a Harris corner interest point detector, and a 300 word vocabulary is learned from a random subset of the extracted features. All SIFT features are thereafter classified as visual words in this vocabulary.

A random frame is selected from each object hypothesis i, and the histogram $\mathbf{a}_i$ of visual words from that frame are used to represent the object. Using the intersection kernel in Eq. (1), the pairwise visual similarity between two objects $\mathbf{a}_1$ and $\mathbf{a}_2$ is defined as $k_{\mathbf{intersect}}(\mathbf{a}_1,\mathbf{a}_2)$.

Fig. 7(c) shows the corresponding pairwise similarities between all appearance features $\mathbf{a}_1$ and $\mathbf{a}_2$ in the training set. There **Containers** cluster is clearer clusters, but the **Tools**, **Ingredients** and **Support areas**, that display fewer



(a) Pairwise string similarities ($K = 3$) (b) Pairwise string histogram similarities (c) Pairwise SIFT BoW similarities

Fig. 7. Object hypothesis similarity statistics. (a) Pairwise similarities $k_{\textbf{string}}(\mathbf{x}_1, \mathbf{x}_2)$ between the object string descriptors, measured with the string kernel, our proposed approach, with $K = 3$. (b) Pairwise similarities $k_{\textbf{intersect}}(\mathbf{h}_1, \mathbf{h}_2)$ between the histograms of object string descriptors, measured with the intersection kernel, baseline 1. (c) Pairwise similarities $k_{\textbf{intersect}}(\mathbf{a}_1, \mathbf{a}_2)$ between the histogram of object SIFT BoW descriptors, measured with the intersection kernel, baseline 2.

appearance characteristics and a higher intra-class variance in terms of appearance, are less clustered in this feature space.

One could build a more sophisticated appearance model that reached a higher performance on this dataset [8] using, e.g., parts based models or a model that takes into account several of the available frames. However, the point of this baseline is to exemplify our thesis that purely appearance-based models do not represent affordances/function per se – they instead appearance properties, like shape and color, which may or may not be correlated to appearance. What is notable in this experiment is that one of the classes **Containers**, have a characteristic appearance, while the others, **Tools**, **Ingredients** and **Support areas** do not. This reflect the tendency of objects to suffer a change in appearance due to human interaction.

d) Classification using the functional and appearance similarity measures: To quantify the results above, the object hypotheses were classified using Support Vector Machines (SVM) [40] with the three kernels $k_{\textbf{string}}(\mathbf{x}_1, \mathbf{x}_2)$ (our proposed method), $k_{\textbf{intersect}}(\mathbf{h}_1, \mathbf{h}_2)$ (baseline 1), and $k_{\textbf{intersect}}(\mathbf{a}_1, \mathbf{a}_2)$ (baseline 2).

The set of object hypotheses described in *a) Dataset* is randomly divided into a training set and a test set of equal size. The same object instances are not present in the training and test sets at the same time. SVMs are then trained with the three similarity measures on the training set, and evaluated using the same similarity measures in the test set. This procedure is executed 500 times and the average classification accuracy is computed.

The parameter K in the string kernel is evaluated in terms of average classification rate. For $K = 2$, the classification rate is 0.90, for $K = 3$ the rate is 0.91, for $K = 5$ it is 0.90. Therefore $K = 3$ is used in all subsequent experiments.

The resulting confusion matrices shown in Fig. 8 support the findings above: The classification accuracy using the functional features ranges from 88% for ingredients to 96% for containers, with an average success rate of 92%. The examples in Fig. 9 give an insight into causes

	Tools	**Ingredients**	**Support areas**	**Containers**
Tools	**0.93**	0.04	0.00	0.03
Ingredients	0.10	**0.88**	0.02	0.00
Support areas	0.00	0.07	**0.90**	0.03
Containers	0.01	0.00	0.02	**0.96**

(a) SVM with $k_{\textbf{string}}(\mathbf{x}_1, \mathbf{x}_2)$, average 92% correct

	Tools	**Ingredients**	**Support areas**	**Containers**
Tools	**0.93**	0.05	0.00	0.02
Ingredients	0.11	**0.82**	0.07	0.00
Support areas	0.00	0.12	**0.85**	0.03
Containers	0.07	0.00	0.01	**0.92**

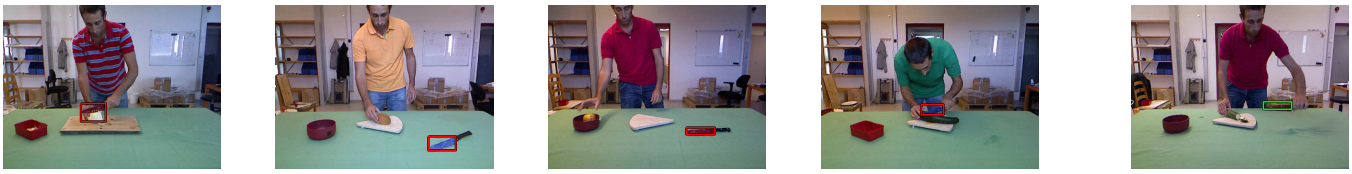
(b) SVM with $k_{\textbf{intersect}}(\mathbf{h}_1, \mathbf{h}_2)$, average 88% correct

	Tools	**Ingredients**	**Support areas**	**Containers**
Tools	**0.60**	0.34	0.02	0.04
Ingredients	0.22	**0.52**	0.19	0.07
Support areas	0.04	0.29	**0.62**	0.05
Containers	0.02	0.10	0.04	**0.84**

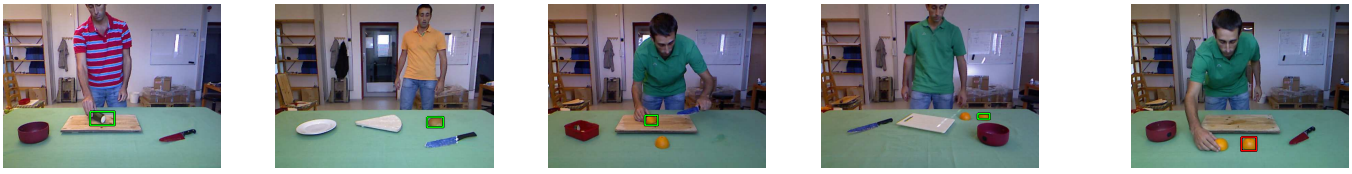
(c) SVM with $k_{\textbf{intersect}}(\mathbf{a}_1, \mathbf{a}_2)$, average 64% correct

Fig. 8. Classification performance. (a) SVM with $k_{\textbf{string}}(\mathbf{x}_1, \mathbf{x}_2)$, our proposed approach, with $K = 3$. (b) SVM with $k_{\textbf{intersect}}(\mathbf{h}_1, \mathbf{h}_2)$, baseline 1. (c) SVM with $k_{\textbf{intersect}}(\mathbf{a}_1, \mathbf{a}_2)$, baseline 2.

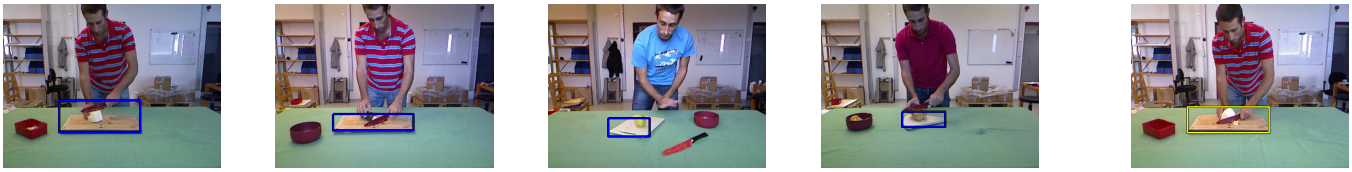
of misclassifications. Tools and ingredients are sometimes confused (Figures 9(b), 9(d)), due to tracking errors which cause ingredient tracks to catch onto the tool interacting with the ingredient and vice versa. These misclassifications are addressed by improving the tracking. Support areas and containers are also sometimes confused (Figures 9(b), 9(d)) for a completely different reason: The hand is sometimes far away from the cutting board when interacting with it (dependent on the type of tool), which causes the interaction to be mislabeled as **Idle**. Similarly, the hand is sometimes near the container without interacting with it, causing an erroneous **Close to** label. These confusions can be disambiguated by modeling object-object interaction in addition to human-object interaction (Section VI-A).



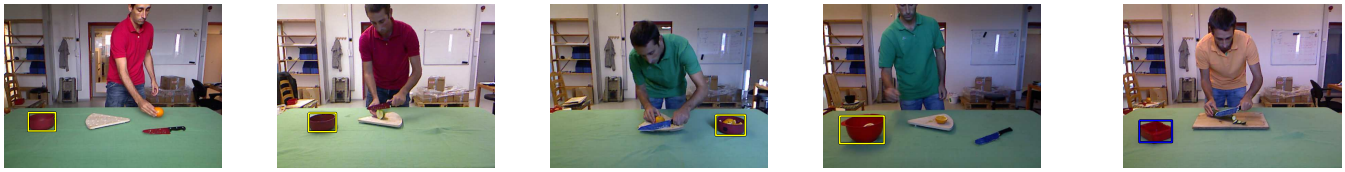
(a) **Tools**, correctly classified (b) misclassified



(c) **Ingredients**, correctly classified (d) misclassified



(e) **Support areas**, correctly classified (f) misclassified



(g) **Containers**, correctly classified (h) misclassified

Fig. 9. Examples of classification results using the functional features. Bounding box color coding: • = **Tools**, • = **Ingredients**, • = **Support areas**, • = **Containers**.

VI. CONCLUSIONS

We present a functional object descriptor, suitable for modeling of human activity, where objects are represented in terms of their interaction with human hands over time. Object hypotheses are first extracted from a video sequence as tracks of associated image segments, and then represented as strings from a 5 word vocabulary, where each word is the type of object-hand interaction in that frame.

Object strings are compared using the string kernel, developed for text retrieval. The string representation together with the kernel constitute a functional representation of objects.

This representation is evaluated on RGB-D video of kitchen scenarios, where a number of different tools, ingredients, containers and cutting boards are involved in human activity in a natural manner. Classification using the proposed functional descriptor had 92% of accuracy on this dataset, performing 28% better than a standard appearance-based representation.

The appearance-based representation performs significantly worse for classes which are in frequent use compared to our approach. This is not surprising as the segmentation of the object hypothesis will change during manipulation generating a range of different appearances which the classifier have to try and generalize over. This means that in order to successfully represent the data using an approach one would need to have examples of every possible variation of the objects for training, image segmentation being a very challenging task this is not a realistic requirement. We can see that for our approach there is no such trend and performance seems to have very little correlation to the activity of the object.

A. *FUTURE WORK*

The functional object descriptors are intended to be used as components of an activity model for robot learning from demonstration. Affordance based, or functional, object representations are also relevant for methods for robot reasoning and planning [41]. We intend to investigate such applications in the future.

A natural extension to the present representation is to learn classifiers that take into account both a similarity in terms of function (the measure presented here) and in terms of appearance (using some sort of appearance feature).

A robot making use of activity models learned from demonstration must also be able to find new instances of the learned functional classes. For this, a model of appearance of each functional class is needed. The training of such a class-specific object detector is a significantly simpler problem than learning the functional multi-class recognizer. Moreover, online refinement of the one-class detector is possible, as the functional recognition method described here will find new training instances every time an activity is observed.

In the near future, the present method will be extended in two ways. Firstly, object-object interaction [34], [32] will be included in the representation along with hand-object interaction. Secondly, we intend to explore methods to learn a symbolic relational abstraction of the data extracted by the segmentation framework such that these symbols allow the learning of abstract transition models rather than, as presently, using a predefined vocabulary of hand-object interaction.

REFERENCES

[1] A. Billard, S. Calinon, R. Dillman, and S. Schaal. Robot programming by demonstration. In B. Siciliano and O. Khatib, editors, *Handbook of Robotics*, chapter 59. Springer, 2008.

[2] S. Schaal. Is imitation learning the route to humanoid robots? *Trends in Cognitive Sciences*, 3:233–242, 1999.

[3] L. Montesano, M. Lopes, A. Bernardino, and J. Santos-Victor. Learning object affordances: From sensory motor coordination to imitation. *IEEE Transactions on Robotics*, 24(1):15–26, 2008.

[4] H. Kjellström. Contextual action recognition. In T. B. Moeslund, A. Hilton, V. Krüger, and L. Sigal, editors, *Guide to Visual Analysis of Humans: Looking at People*, chapter 18. Springer, 2011.

[5] A. Gupta, A. Kembhavi, and L. S. Davis. Observing human-object interactions: Using spatial and functional compatibility for recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(10):1775–1789, 2009.

[6] H. Kjellström, J. Romero, and D. Kragic. Visual object-action recognition: Inferring object affordances from human demonstration. *Computer Vision and Image Understanding*, 115(1):81–90, 2011.

[7] H. Pirsiavash and D. Ramanan. Detecting activities of daily living in first-person camera views. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2012.

[8] L. Fei-Fei, R. Fergus, and A. Torralba. *Recognizing and Learning Object Categories: Short course at ICCV*, 2009.

[9] J. J. Gibson. *The Ecological Approach to Visual Perception*. Lawrence Erlbaum Associates, 1979.

[10] E. Rivlin, S. J. Dickinson, and A. Rosenfeld. Recognition by functional parts. *Computer Vision and Image Understanding*, 62(2):164–176, 1995.

[11] M. A. Sutton and L. Stark. Function-based reasoning for goal-oriented image segmentation. In B. Siciliano and O. Khatib, editors, *Handbook of Robotics*, chapter 59. Springer, 2008.

[12] J. Gall, A. Fossati, and L. van Gool. Functional categorization of objects using real-time markerless motion capture. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2011.

[13] M. Stark, P. Lies, M. Zillich, J. Wyatt, and B. Schiele. Functional object class detection based on learned affordance cues. In *International Conference on Computer Vision Systems*, 2008.

[14] N. Bergström, C. H. Ek, M. Björkman, and D. Kragic. Scene understanding through interactive perception. In *International Conference on Computer Vision Systems*, 2011.

[15] T. Hermans, J. M. Rehg, and A. Bobick. Guided pushing for object singulation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012.

[16] H. Grabner, J. Gall, and L. van Gool. What makes a chair a chair? In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2011.

[17] V. Delaitre, D. F. Fouhey, I. Laptev, J. Sivic, A. Gupta, and A. A. Efros. Scene semantics from long-term observation of people. In *European Conference on Computer Vision*, 2012.

[18] A. Gupta, S. Satkin, A. A. Efros, and M. Hebert. From 3D scene geometry to human workspace. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2011.

[19] M. W. Turek, A. Hoggs, and R. Collins. Unsupervised learning of functional categories in video scenes. In *European Conference on Computer Vision*, 2010.

[20] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2011.

[21] H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. Watkins. Text classification using string kernels. *Journal of Machine Learning Research*, 2:419–444, 2002.

[22] J. Bohg and D. Kragic. Grasping familiar objects using shape context. In *International Conference on Advanced Robotics*, 2009.

[23] M. Madry, D. Song, and D. Kragic. From object categories to grasp transfer using probabilistic reasoning. In *IEEE International Conference on Robotics and Automation*, 2012.

[24] A. Saxena, J. Driemeyer, and A. Y. Ng. Robotic grasping of novel objects using vision. *International Journal of Robotics Research*, 27(2):157–173, 2008.

[25] R. Filipovych and E. Ribeiro. Recognizing primitive interactions by exploring actor-object states. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2008.

[26] D. J. Moore, I. A. Essa, and M. H. Hayes. Exploiting human actions and object context for recognition tasks. In *IEEE International Conference on Computer Vision*, 1999.

[27] P. Peursum, G. West, and S. Venkatesh. Combining image regions and human activity for indirect object recognition in indoor wide-angle views. In *IEEE International Conference on Computer Vision*, 2005.

[28] M. Veloso, F. von Hundelshausen, and P. E. Rybski. Learning visual object definitions by observing human activities. In *IEEE-RAS International Conference on Humanoid Robots*, 2005.

[29] J. Wu, A. Osuntogun, T. Choudhury, M. Philipose, and J. M. Rehg. A scalable approach to activity recognition based on object use. In *IEEE International Conference on Computer Vision*, 2007.

[30] P. Fitzpatrick, G. Metta, L. Natale, S. Rao, and G. Sandini. Learning about objects through action – initial steps towards artificial cognition. In *IEEE International Conference on Robotics and Automation*, 2003.

[31] L. Stark and K. Bowyer. *Generic Object Recognition using Form and Function*. World Scientific Series in Machine Perception and Artificial Intelligence – Vol. 10, 1996.

[32] B. Moldovan, P. Moreno, M. van Otterlo, J. Santos-Victor, and L. De Raedt. Learning relational affordance models for robots in multi-object manipulation tasks. In *IEEE International Conference on Robotics and Automation*, 2012.

[33] E. E. Aksoy, A. Abramov, F. Wörgötter, and B. Dellen. Categorizing object-action relations from semantic scene graphs. In *IEEE International Conference on Robotics and Automation*, 2010.

[34] G. Luo, N. Bergström, C. H. Ek, and D. Kragic. Representing actions with kernels. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011.

[35] L. Nalpantidis, G. Sirakoulis, and A. Gasteratos. Non-probabilistic cellular automata-enhanced stereo vision simultaneous localization and mapping. *Measurement Science and Technology*, 22(11), 2011.

[36] J. Matas, C. Galambos, and J. Kittler. Robust detection of lines using the progressive probabilistic Hough transform. *Computer Vision and Image Understanding*, 78:119–137, 2000.

[37] L. Shapiro and G. Stockman. Connected components labeling. In Prentice Hall, editor, *Computer Vision*, chapter 3, pages 69–73. Prentice Hall, 2002.

[38] J.-Y. Bouguet. Pyramidal implementation of the Lucas Kanade Feature Tracker, description of the algorithm, 2000.

[39] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[40] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines*. Cambridge University Press, 2000.

[41] T. Lang and M. Toussaint. Planning with noisy probabilistic relational rules. *Journal of Artificial Intelligence Research*, 39:1–49, 2010.